REMARKS

Reconsideration of this application, as amended, is respectfully requested.

Claims 1-9, 11-13, 15-33, 35-37, and 39-50 are pending. Claims 1-9, 11-13, 15-33, 35-37, and 39-50 have been rejected. Claims 11 and 35 have been objected to.

Claims 1, 4, 5, 11, 25, 28, 29, and 35 have been amended. Claims 2 and 26 have been canceled. No claims have been added. Support for the amendments is found in the specification, the drawings, and in the claims as originally filed. Applicants submit that the amendments do not add new matter.

Applicants reserve all rights with respect to the applicability of the Doctrine of Equivalents.

Examiner objected to claims 11 and 35 because of informalities.

Applicants have amended claims 11 and 35.

Therefore, applicants submit that the Examiner's objections with respect to claims 11 and 35 have been overcome.

Claims 1-9, 25-33, and 49 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,446,198 to Sazegari (hereinafter "Sazegari") in view of U.S. Patent No. 6,397,324 to Barry et al. (hereinafter "Barry") and further in view of U.S. Patent No. 5,768,628 to Priem (hereinafter "Priem").

Claim 1 reads as follows:

> A method for execution by a microprocessor in response to receiving a single instruction, the method comprising:
> receiving the single instruction that includes a first index of a first vector in a first entry in a register file and a second index of a second vector in a second entry in the register file, receiving the first vector having a first plurality of numbers and the second vector having a second plurality of numbers, each of the first plurality of numbers pointing to one of a

> plurality of entries, each of the plurality of entries being in one of a plurality of look-up tables; and
> replacing simultaneously the plurality of entries in the plurality of look-up tables that are indicated by the first plurality of numbers from the first entry in the register file, with the second plurality of numbers from the second entry in the register file;
> wherein the receiving and the replacing operations are performed in response to the microprocessor receiving the single instruction;
> wherein the microprocessor comprises a media processor integrated with a memory controller for host memory on a single integrated circuit.

(Amended claim 1) (emphasis added)

It is respectfully submitted that Sazegari does not teach or suggest a combination with Barry and Priem, Barry does not teach or suggest a combination with Sazegari and Priem, and Priem does not teach or suggest a combination with Sazegari and Barry. It would be impermissible hindsight, based on applicants' own disclosure, to combine Sazegari, Barry, and Priem.

The Examiner acknowledged that "Sazegari has not taught the operation being replacing" (Office Action, p. 4, 9/19/07).

Sazegari discloses a vectorized table lookup. More specifically, Sazegari discloses:

> In a vectorized processing unit, one approach that has been employed to perform multiple simultaneous table lookups is through the use of the "permute" instruction. This instruction operates to fill a register with data values from two other registers. The data values can be specified in any order. Referring to FIG. 3, a permute mask is stored in a register 26, and values that are to be used to form the final result are stored in two data registers 28 and 30. The permute instruction uses the mask values in the register 26 to assign corresponding values of the operands in the registers 28 and 30 to a result register 32. In the illustrated example, byte(1) of register 28 is mapped to byte(0) of the result register, and byte(14) of register 30 is mapped to byte(1) of the result register. In this example, each of the registers stores 16 bytes, i.e. 128 bits.

(Sazegari, col. 4, lines 5-19) (emphasis added)

In particular, Sazegari discloses:

> For table lookup operations, the permute instruction can be used to perform 16 simultaneous lookup operations on a 32-byte entry table. FIG. 4 illustrates such a table 34, which consists of two 16-byte vectors, data1 and data2. Each vector can be stored in one register of the CPU. The permute instruction can be used to simultaneously read 16 values from these two vectors, in accordance with index values in a register 36, and store the 16 output results in sequential order in another register 38.

(Sazegari, col. 4, lines 24-32) (emphasis added)

Thus, Sazegari discloses the permute instruction that is used to read values from the two vectors, in accordance with index values in a register. In contrast, amended claim 1 refers to receiving the single instruction that has a first index of a first vector in a first entry in a register file and a second index of a second vector in a second entry in the register file, receiving the first vector having a first plurality of numbers and the second vector having a second plurality of numbers, each of the first plurality of numbers pointing to one of a plurality of entries, each of the plurality of entries being in one of a plurality of look-up tables; and replacing simultaneously the plurality of entries in the plurality of look-up tables that are indicated by the first plurality of numbers from the first entry in the register file, with the second plurality of numbers from the second entry in the register file.

Barry, in contrast, discloses accessing tables in memory banks using load and store address generators sharing store read port of compute register file separated from address register file (Abstract).

Priem, in contrast, discloses the system memory to store the wave tables (Abstract).

Furthermore, even if the accessing tables in the memory banks of Barry and the system memory of Priem were incorporated into vectorized table lookup of Sazegari, such a combination would still lack receiving the single instruction that includes a first index of a first vector in a first entry in a register file and a second index of a second vector in a second entry in the register file, receiving the first vector having a first plurality of numbers and the second vector having a second plurality of numbers, each of the first plurality of numbers pointing to one of a plurality of entries, each of the plurality of entries being in one of a plurality of look-up tables; and replacing simultaneously the plurality of entries in the plurality of look-up tables that are indicated by the first plurality of numbers from the first entry in the register file, with the second plurality of numbers from the second entry in the register file, as recited in amended claim 1.

Therefore, applicants respectfully submit that amended claim 1 is not obvious under 35 U.S.C. § 103(a) over Sazegari, in view of Barry, and further in view of Priem.

Given that claims 2-3 and 26-27 contain limitations that are similar to those discussed with respect to amended claim 1, applicants respectfully submit that claims 2-3 and 26-27 are not obvious under 35 U.S.C. § 103(a) over Sazegari, in view of Barry, and further in view of Priem.

Amended claim 4 reads as follows:

> A method for execution by a microprocessor in response to receiving a single instruction, the method comprising:
>
> receiving the single instruction having an identity number code that specifies a DMA controller and an index of a first entry in a register file, wherein the first entry in the register file contains control parameters that include a bit segment which specifies a count indicating a number of entries to be loaded in each of a plurality of look-up units;
>
> receiving the control parameters from the register file; and
>
> replacing at least one entry in at least one of the plurality of look-up units in a microprocessor unit according to the control parameters with at least one number using the Direct Memory Access (DMA) controller;
>
> wherein the replacing is performed in response to the microprocessor receiving the single instruction.

(Amended claim 4)(emphasis added)

It is respectfully submitted that Sazegari does not teach or suggest a combination with Barry and Priem, Barry does not teach or suggest a combination with Sazegari and Priem, and Priem does not teach or suggest a combination with Sazegari and Barry. It would be impermissible hindsight, based on applicants' own disclosure, to combine Sazegari, Barry, and Priem.

Sazegari discloses a vectorized table lookup. More specifically, Sazegari discloses that the "permute instruction uses the mask values in the register 26 to assign corresponding values of the operands in the registers 28 and 30 to a result register 32" (Figure 3, col. 4, lines 12-15).

Thus, Sazegari merely discloses that the permute instruction uses values in the register to assign values of the operands. In contrast, amended claim 4 refers to receiving the single instruction having an identity number code that specifies a DMA controller and an index of a first entry in a register file, wherein the first entry in the register file contains control parameters that include a bit segment which specifies a count indicating a number [how many] of entries to be loaded in each of a plurality of look-up units.

Barry, in contrast, discloses accessing tables in memory banks using load and store address generators sharing store read port of compute register file separated from address register file (Abstract).

Priem, in contrast, discloses the system memory to store the wave tables (Abstract).

Furthermore, even if the accessing tables in the memory banks of Barry and the system memory of Priem were incorporated into vectorized table lookup of Sazegari, such a combination would still lack receiving the single instruction having an identity number code that specifies a DMA controller and an index of a first entry in a register file, wherein the first entry in the register file contains control parameters that include a bit segment which specifies a count indicating a number [how many] of entries to be loaded in each of a plurality of look-up units, as recited in amended claim 4.

Therefore, applicants respectfully submit that amended claim 4 is not obvious under 35 U.S.C. § 103(a) over Sazegari, in view of Barry.

Given that claims 5-9, 28-33, and 49 contain limitations that are similar to those discussed with respect to amended claim 4, applicants respectfully submit that claims 5-9, 28-33, and 49 are not obvious under 35 U.S.C. § 103(a) over Sazegari, in view of Barry.

Claims 11-24 and 35-48 are rejected under 35 U.S.C. § 103(a) as being taught by Sazegari in view of Barry.

Amended claim 11 reads as follows:

> A method for execution by a microprocessor in response to receiving a single instruction, the method comprising:
>
> receiving the single instruction that includes a first index of a first vector in a first entry in a register file and a second index of a second vector in a second entry in the register file;
>
> receiving the first vector having a plurality of numbers from the register file;
>
> partitioning a look-up memory into a first plurality of look-up tables, wherein the look-up memory comprises a second plurality of look-up units, and wherein the partitioning of the look-up memory comprises configuring the second plurality of look-up units into the first plurality of look-up tables according to a configuration indicator specified by the single instruction;
>
> looking up simultaneously a plurality of elements of the second vector from the first plurality of look-up tables, each of the plurality of elements being in one of the first plurality of look-up tables and being pointed to by one of the plurality of numbers;
>
> wherein the partitioning and the looking-up operations are performed in response to the microprocessor receiving the single instruction .

(Amended claim 11)(emphasis added)

It is respectfully submitted that Sazegari does not teach or suggest a combination with Barry, and Barry does not teach or suggest a combination with Sazegari. It would be impermissible hindsight, based on applicants' own disclosure, to combine Sazegari and Barry.

Sazegari discloses a vectorized table lookup (Abstract) More specifically, Sazegari discloses that the permute instruction is used to read values from the two vectors, in accordance with index values in a register (col. 2, lines 17-43, col. 4, lines 5-19, 26-32, 41-46, and 59-63, Figures 2, 4, and 6). In contrast, amended claim 11 refers to receiving the single instruction that includes receiving the single instruction that includes a first index of a first vector in a first entry in a register file and a second index of a second vector in a second entry in the register file; receiving the first vector having a plurality of numbers from the register file; and looking up simultaneously a plurality of elements of the second vector [indicated by the single instruction] from the first plurality of look-up tables, each of the plurality of elements being in one of the first plurality of look-up tables and being pointed to by one of the plurality of numbers [of the first vectors from the register file indicated by the single instruction].

Barry, in contrast, discloses accessing tables in memory banks using load and store address generators sharing store read port of compute register file separated from address register file (Abstract). More specifically, Barry discloses load and store table instructions (col. 10, lines 23-col. 11, line 65). In particular, Barry discloses "… instruction can be used to simultaneously load two different values from any data-dependent memory address in each local memory bank" (col. 11, lines 29-35). In contrast, amended claim 11 refers to receiving the single instruction that includes receiving the single instruction that includes a first index of a first vector in a first entry in a register file and a second index of a second vector in a second entry in the register file; receiving the first vector having a plurality of numbers from the register file; and looking up simultaneously a plurality of elements of the second vector [indicated by the single instruction] from the first plurality of look-up tables, each of the plurality of elements being in one of the first plurality of look-up tables and being pointed to by one of the plurality of numbers [of the first vectors from the register file indicated by the single instruction].

Furthermore, even if the accessing tables in the memory banks of Barry were incorporated into vectorized table lookup of Sazegari, such a combination would still lack receiving the single instruction that includes receiving the single instruction that includes a first index of a first vector in a first entry in a register file and a second index of a second vector in a second entry in the register file; receiving the first vector having a plurality of numbers from the register file; and looking up simultaneously a plurality of elements of the second vector [indicated by the single instruction] from the first plurality of look-up tables, each of the plurality of elements being in one of the first plurality of look-up tables and being pointed to by one of the plurality of numbers [of the first vectors from the register file indicated by the single instruction], as recited in amended claim 11.

Therefore, applicants respectfully submit that amended claim 11 is not obvious under 35 U.S.C. § 103(a) over Sazegari, in view of Barry.

Given that claims 12-13, 15-24, 35-37, and 39-48 contain limitations that are similar to those discussed with respect to amended claim 11, applicants respectfully submit that claims 12-13, 15-24, 35-37, and 39-48 are not obvious under 35 U.S.C. § 103(a) over Sazegari, in view of Barry.

Claim 50 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Sazegari in view of Barry, as applied to claim 11 above, and further in view of Priem.

As set forth above, Sazegari does not teach or suggest a combination with Barry and Priem, Barry does not teach or suggest a combination with Sazegari and Priem, and Priem does not teach or suggest a combination with Sazegari and Barry. It would be impermissible hindsight, based on applicants' own disclosure, to combine Sazegari, Barry, and Priem.

Furthermore, even if Barry and Priem were incorporated into Sazegari, such a combination would still lack receiving the single instruction that includes receiving the single instruction that includes a first index of a first vector in a first entry in a register file and a second index of a second vector in a second entry in the register file; receiving the first vector having a plurality of numbers from the register file; and looking up simultaneously a plurality of elements of the second vector [indicated by the single instruction] from the first plurality of look-up tables, each of the plurality of elements being in one of the first plurality of look-up tables and being pointed to by one of the plurality of numbers [of the first vectors from the register file indicated by the single instruction], as recited in amended claim 11.

Given that claim 50 depends from amended claim 11, and add additional limitations, applicants respectfully submit that claim 50 is not obvious under 35 U.S.C. § 103(a) over Sazegari, in view of Barry, and further in view of Priem.

It is respectfully submitted that in view of the amendments and arguments set forth herein, the applicable rejections and objections have been overcome. If there are any additional charges, please charge Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: 01/18/2008

/Tatiana Rossin/
Tatiana Rossin
Reg. No. 56,833

1279 Oakmead Parkway
Sunnyvale, CA 94085-4040
Phone: (408) 720-8300
Fax: (408) 720-8383